Boeing Defense, Space & Security
Lean-Agile Software

An Integrated Toolset for Agile Systems Engineering Requirements Analysis

Phyllis Marbach

19 May 2011

This document does not contain technical data within the definition contained in the International Traffic in Arms Regulations (ITAR) and the Export Administration Regulations (EAR), as such is releasable by any means to any person whether in the U.S. or abroad. The Export Compliance log number for this document is Export Approval # RBE3973-NT (assigned IAW PRO-4527, PRO 3439).

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
19 MAY 2011		**00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
An Integrated Toolset for Agile Systems Engineering Requirements Analysis	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Boeing Defense, Space & Security,PO Box 516,St. Louis,MO,63166**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Presented at the 23rd Systems and Software Technology Conference (SSTC), 16-19 May 2011, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License

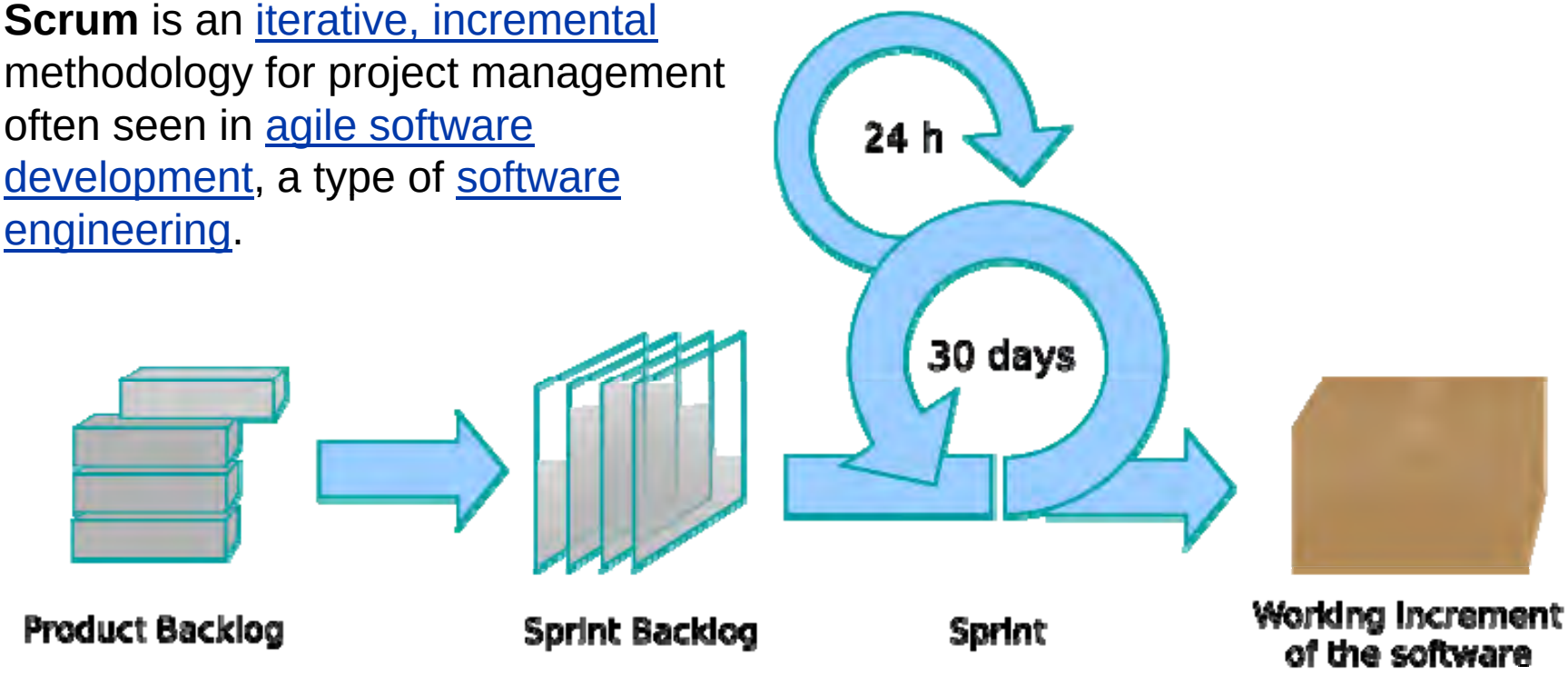
14. ABSTRACT
An Integrated Toolset for Agile Systems Engineering Requirements Analysis introduces an iterative approach to analyzing requirements and the integrated toolset that enables this analysis. The requirements analysis may occur in parallel and one iteration ahead of software development or it may occur after software exists to produce needed artifacts as in the case to be presented. An agile process for systems engineering to analyze requirements and provide to the agile software team will be introduced. The commercial-off-the-shelf tools that allow collaboration, standard formatting, logging unique identification numbering, linking to defined tests, and formal document production, configuration control and actual program use will be presented. This end-to-end integrated toolset allows the sync-up of the requirements to test descriptions not only at each formal release, but daily as the iterations Copyright ? 2010 Boeing. All rights reserved. 28 requirements evolve during the iterations. This results in planned deliverable products at the end of each iteration to internal and external customers.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**28**	

Introduction to Agile (Scrum)

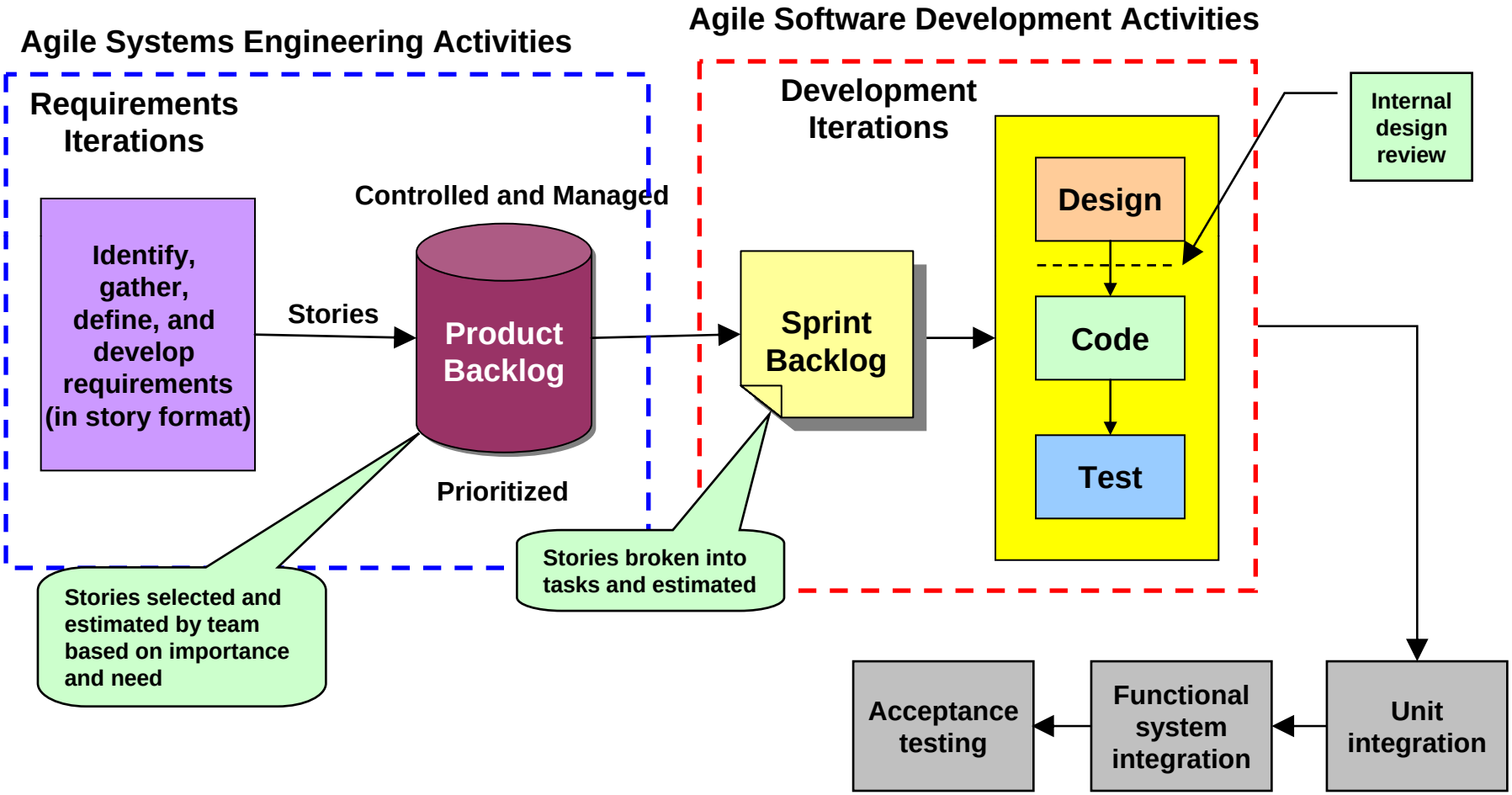
Scrum is an iterative, incremental methodology for project management often seen in agile software development, a type of software engineering.



Copyrights specified as freely licensed media
http://en.wikipedia.org/wiki/File:Scrum_process.svg

Introduction to Agile Systems Engineering

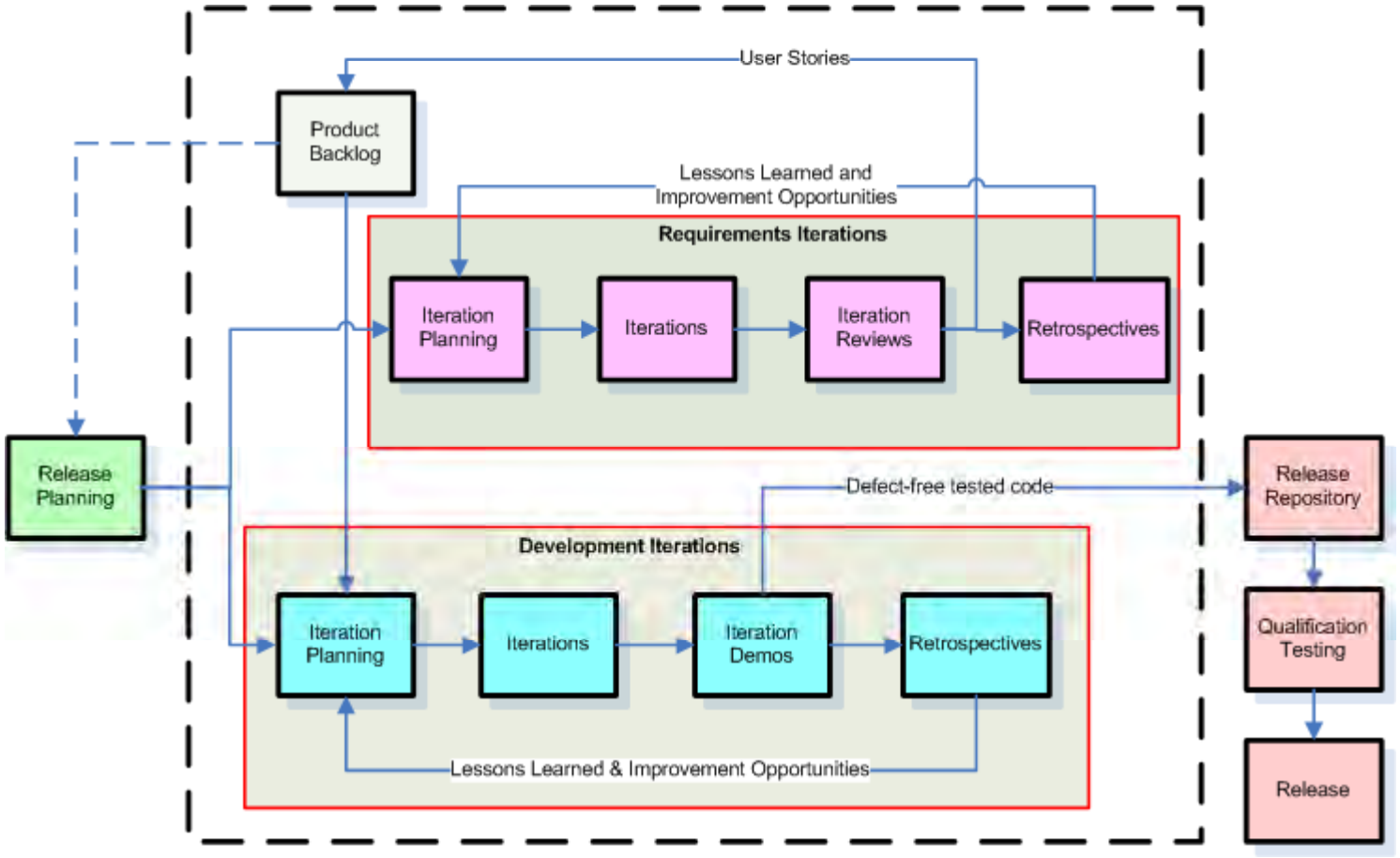
For Software Development Requirements Analysis



Richard Carlson, Phillip Matzuc; A Viable Systems Engineering Approach, SSTC 2010

Agile Sys Engrg Requirements Analysis

Parallel and One Iteration Ahead of Software Development

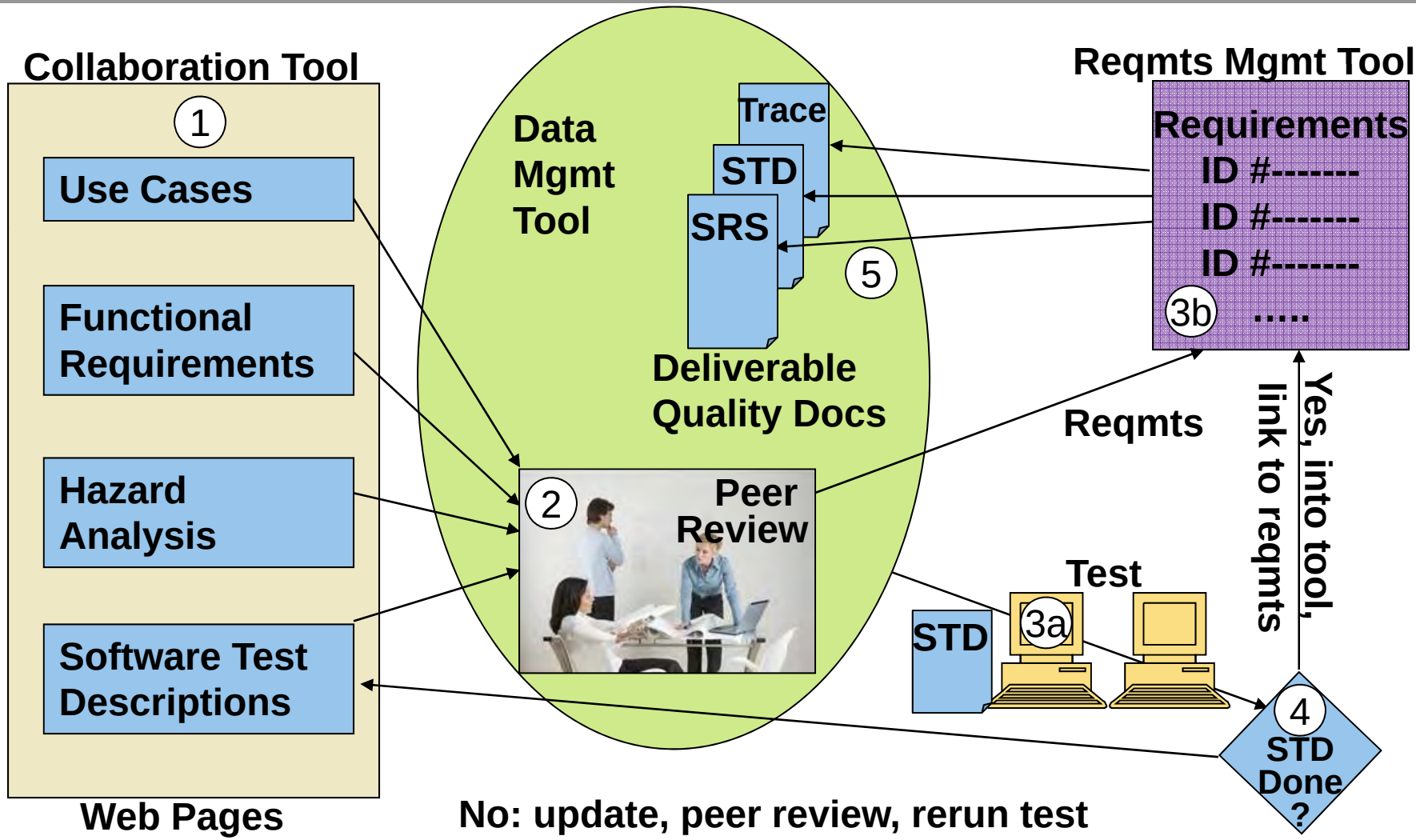


Richard Carlson, Phillip Matzuc; A Viable Systems Engineering Approach, SSTC 2010

Analysis of Existing Code

- **Have:**
  - Code
  - Test Bed
  - User Interface
  - User Procedures
- **Lack:**
  - Requirements documentation
  - Architecture & Design Diagrams
  - Trace Matrix of Tests to Requirements
  - Software Test Descriptions
  - Hazard Analysis

Analysis with Integrated Toolset



Getting Started

- **Code**
- **Domain experts – not always available**
- **Existing documentation in program repositories – charts, operator procedures**
- **Determine Next Steps**

Epics and Backlog Items

- **30 Epics were created from the User Interface Features, examples:**
  - Power On
  - Start Up Feature
  - Shutdown Feature
  - Operate Component
  - Operate Another Component
- **Product Owner prioritized the most important ones**
- **Each epic has 5 significant backlog items (took 3 Iterations to reach these 5):**
  - Functional Analysis
  - Requirements
  - Hazard Analysis
  - Draft Test Procedure
  - Finalize Test Procedure

Create Documentation Feature by Feature

- **Created templates**
- **Goal is to identify tasks that take 16 hour max**
- **Include what "Done" means in the template**

Backlog Item Templates

Filter

Move to Project

Title	ID
User Story Template - OLD	B-01017
Task Template - OLD	B-01018
Update Documentation or Work Products Template	B-01243
Functional Analysis Template	B-01225
Research and Document Functionality	TK-02479
Requirements Template	B-01216
Generate Functional Requirements	TK-02420
Peer Review Requirements	TK-02422
Update and Post Requirements	TK-02469
Hazard Analysis Template	B-01219
Identify and Analyze Potential Hazards	TK-02546
Peer Review Hazard Analysis	TK-02547
Update Hazard Analysis	TK-02548
Draft Test Procedures Template	B-01252
Generate Draft Test Procedures	TK-02543
Peer Review Draft Test Procedures	TK-02544
Update and Post Draft Test Procedures	TK-02545
Finalize Test Procedure Template	B-01251
Run Test Procedures	TK-02539
Update and Post Finalized Test Procedures	TK-02540

Manage the Backlog

- **Application Lifecycle Management (ALM) Tools**
  - IBM – Rational Team Concert
  - MKS, Inc.
  - Atlassian - JIRA with GreenHopper
  - CollabNet
  - HP
  - Micro Focus
  - Microsoft
  - Rally Software Development
  - Serena Software
  - VersionOne

http://adtmag.com/articles/2010/05/12/ibm-mks-have-best-agile-management-tools.aspx

Integrated End-to-End Toolset

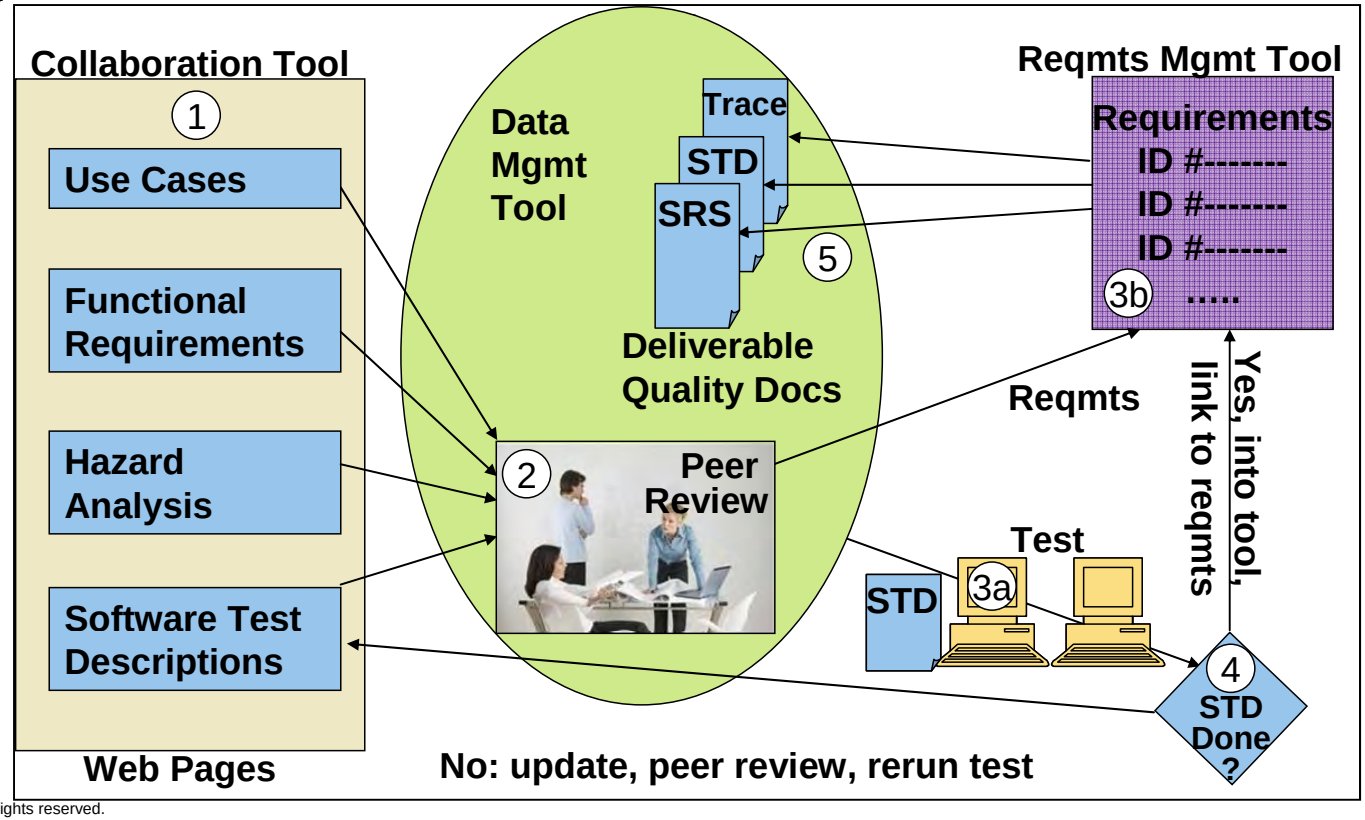
Collaboration

Standard formatting

Data Management

Configuration Control

Standard formatting
Logging
Unique identification numbering
Linking to defined tests
Formal document production



Collaboration

- **Collaboration tool should be:**
  - Easy to access
  - Easy to use
  - Easy to comment
  - Easy to change
- **Team started with a collaborative tool**
  - Mediawiki, open source
  - TWiki™, open source, collaboration platform
  - Confluence
  - SharePoint
  - Socialtext

Collaboration – Home Page

- **Introduction about the analysis underway**
- **Link to a list of functional threads: links have the work products themselves**
- **Links to references used**
- **Links to test environment information**
- **Links to templates for work products with instructions**
  - Collaboration Tool Templates
  - Functional Descriptions
  - Requirements/Use Cases/
  - Hazard Analysis/Risk Mitigation
  - Test Procedures/Test Cases/Test Descriptions

Example

- **Collaboration tool exports content to a Word Document**
- **Word Document is parsed into DOORS**
- **All feature reqmts in DOORS create final SRS**
- **Released documents are baselined in the Data Mgmt Tool Repository that provides Configuration Mgmt control**

- Description of Functionality
  - Overview
  - Functional Decomposition
  - Use Case Development
    - Phase 1 level

- Requirements
  - Use Case Development
    - Phase 1 (operator/functional) level
  - Functional Requirements
  - Requirements Documents
    - SRS Document in TWiki
    - SRS Document in DOORS
    - SRS Document in PIMS

- Test Procedures
  - Existing Test Procedures
  - FQT Team Test Case/Test Procedure Development
    - Test Cases
    - Test Procedure Document
    - Expected Test Results
    - Test Procedures to Requirements Trace
  - Software Test Description (STD)

- Test Results

- Hazard Analysis/Risk Mitigation
  - Hazards/Mitigation

Data Management Tool / Repository

- **Capabilities include:**
  - Draft folders/repository
  - Peer Review records
  - Action Item creating/tracking/closure
  - Release folders/repository
  - Calendar
  - Meeting notification
  - Distribution Lists and access control to records
  - Configuration Management work flow and approvals
  - Collaboration across companies, subcontractors, customers
- **Examples (to name a few)**
  - Master Data Management Tool: Microsoft, Data Foundations, Kalido,
  - Business to Business Tools, Amalto Technologies, Entreon Corp.,

Documentation

- **During each iteration:**
  - Software Requirement Specification is created Feature by Feature rather than all at once.
  - Software Test Descriptions are created as each feature is analyzed
  - Hazard Analysis is performed one feature at a time.
- **At each release:**
  - More features are complete within the SRS,
  - More STDs are complete and
  - More Hazards Analysis are complete.

Peer Reviews

- **Each backlog item included conducting peer reviews of the content.**
- **The peer review was the acceptance criteria before work could be posted into DOORS or claimed done and be included into the demonstration of the iteration**
- **One team member responsible for the entire backlog item of tasks but other team members might be actually performing the tasks.**
- **Conducting peer reviews, as soon as possible, instead of waiting until the week before the demonstration, helped get findings removed and more work ready for demonstration.**
- **Peer Review records were kept in a data management tool where the data included:**
  - Artifact in review
  - Peer review date,
  - Personnel reviewing,
  - Time spent reviewing,
  - Findings discovered and removed

Define “Done” – Includes Logging

- **Work was not complete until content was posted into the Requirements Management Tool**
- **INCOSE site has 34 listed as of 3/18/2011: http://www.incose.org/ProductsPubs/products/rmsurvey.aspx**
- **Two in use at Boeing:**
  - IBM Rational DOORS
  - IBM Rational RequisitePro
- **Unique record Identification numbers are automatically assigned**

Software Test Descriptions

- **Developed the software test descriptions**
- **Ran these in the Test Lab to verify complete**
- **Found some common repeatable test steps**
- **Created these as common test descriptions that could be called from other procedures reducing work and making future test procedure development faster.**
- **Linking the STDs to the requirements in the requirements management tool began the Trace Matrix**

Formal Document Production

- **The format for the Software Requirement Specification followed company standards and was populated into the requirements management tool**
- **As each release occurred the document produced met format standards.**
- **The format of the Software Test Descriptions followed company standards and was posted into the collaboration tool so all documents started with the standard format**
- **STDs were posted into requirements management tool and content linked to requirements to enable the production of the Trace Matrix eventually.**

Data Availability & Consistency

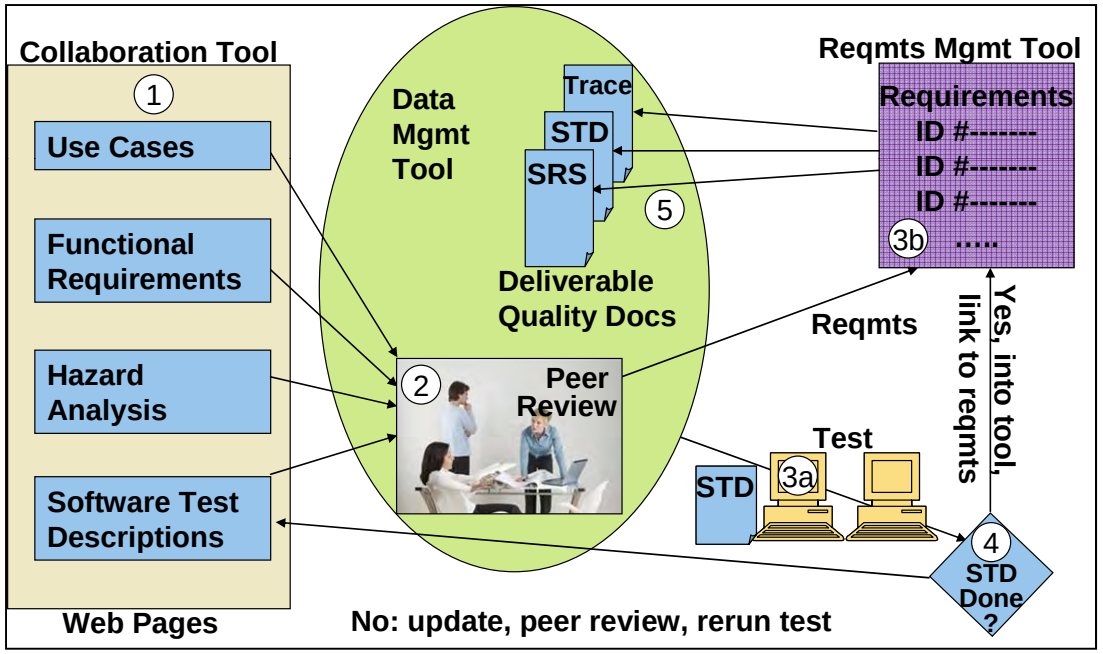
- **Each day the requirements in the Requirements Management Tool was exported into an HTML file onto a web page so that those not familiar with or licensed for the Requirements Management Tool could see the latest, most complete list**
- **Each month the requirements and test descriptions in the Requirements Management Tool could be exported into word files to be posted as drafts into the Data Management Repository so those more comfortable working with documents could see the latest demonstrated list**
- **Each quarter the requirements and test descriptions in the Requirements Management Tool were exported into word files to be posted as releases into the Data Management Repository**

Configuration Control for SRS

- **Sequence of events:**
  - Develop draft in collaboration tool
  - Perform peer review using the data management tool
  - Populate requirements management tool
  - Create the final SRS word document from the requirements management tool and post baseline into the data management tool.
  - Changes to baselined content approved in a Change Board
  - Approved changes added to Product Backlog based on priority

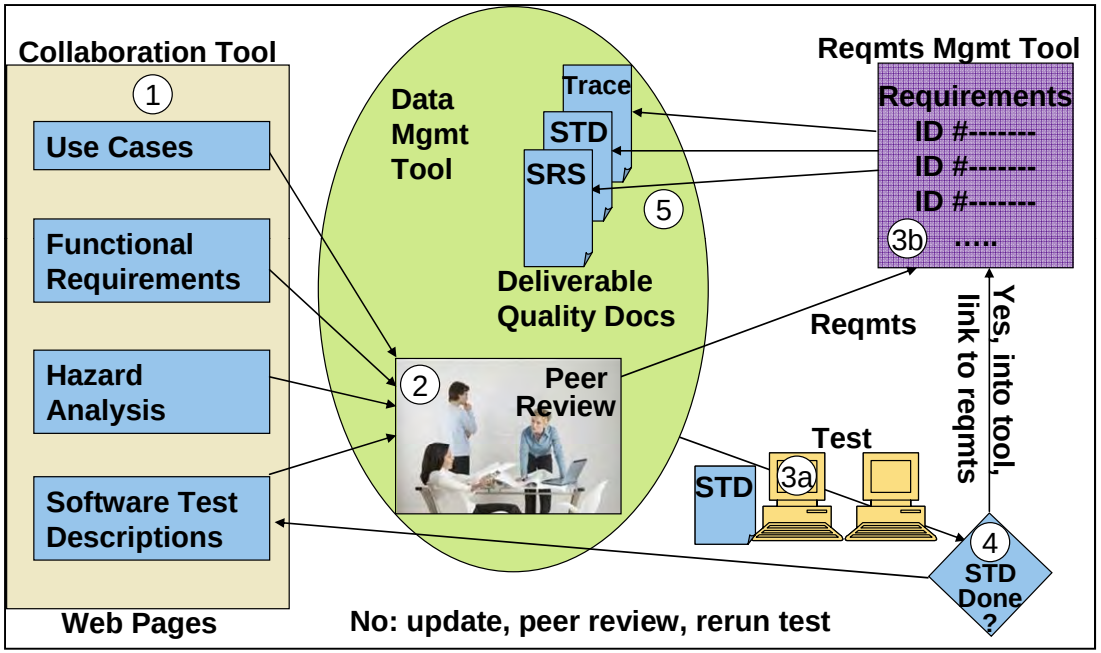
Configuration Control for STD

- **Sequence of events:**
  - Develop draft in collaboration tool
  - Perform peer review using the data management tool
  - Run the test in the test lab, redline the STD
  - Update the collaboration tool with redlines
  - Update the requirements as needed
  - Perform peer review with updates/redlines
  - Run the updated test in the test lab



Configuration Control for STD (Cont.)

- **Sequence of events:**
  - Finalize the STD
  - Populate the STD into the requirements management tool
  - Link the test descriptions to the requirements that are verified
  - Create the final STD word document from the requirements management tool and post baseline into the data management tool.
  - Changes to baselined content approved in a Change Board
  - Approved changes added to Product Backlog based on priority



Agile Practices Drive LEAN Disciplines

LEAN Disciplines	Agile Requirements Analysis
1. Establish Clear Priorities	1. Product backlog is always prioritized; Team works on highest priority items first
2. Eliminate Bad Multitasking – Focus and Finish	2. Team is shielded from interruptions that cause bad multitasking
3. Limit the Release of Work in Process (WIP) to Deliver Earlier	3. Tasks are pulled from the iteration backlog one at a time to limit individual WIP
4. Prepare! Start ⟶ Finish	4. Requirements are not selected from the product backlog until everything needed is available
5. Use Checklists to Prevent Defects and Traveled Risk	5. Checklists and guides are used to prevent costly rework
6. Face into and Resolve Issues Quickly	6. Daily stand-up meetings force issues and risks to be identified and resolved quickly
7. Drive Daily Execution	7. Daily stand-up meetings drive team-based execution

Acronyms and Abbreviations

Docs	Documents
DOORS	Dynamic Object-Oriented Requirements System
EAR	Export Administration Regulations
FQT	Functional Qualification Test
h	Hours
HTML	Hypertext Markup Language
ID	Identification
INCOSE	International Council On Systems Engineering
IAW	In Accordance With
ITAR	International Traffic in Arms Regulation
Mgmt	Management
PRO	Boeing Procedure
Reqmts	Requirements
SRS	Software Requirement Specification
STD	Software Test Description
Sys Engrg	System Engineering

Author Biography

- **Phyllis R. Marbach is a Senior Software Manager in Boeing's Defense Space and Security (BDS). Marbach has over 32 years experience in aerospace programs including Satellites, chemical lasers, the International Space Station, and various propulsion systems. Currently she is a team lead with the Lean-Agile Software Services (LASS) for the BDS LASS Coaching Team , a Boeing Agile Software Process (BASP) Trainer and an active BASP Coach working with Unmanned Air Systems, Radio, and research programs. phyllis.r.marbach@boeing.com**
- **Marbach holds a BS in Chemistry and Applied Mathematics from Centre College of Kentucky and an MS degree in Engineering from UCLA.**

Abstract

An Integrated Toolset for Agile Systems Engineering Requirements Analysis introduces an iterative approach to analyzing requirements and the integrated toolset that enables this analysis. The requirements analysis may occur in parallel and one iteration ahead of software development or it may occur after software exists to produce needed artifacts as in the case to be presented. An agile process for systems engineering to analyze requirements and provide to the agile software team will be introduced. The commercial-off-the-shelf tools that allow collaboration, standard formatting, logging, unique identification numbering, linking to defined tests, and formal document production, configuration control and actual program use will be presented. This end-to-end integrated toolset allows the sync-up of the requirements to test descriptions not only at each formal release, but daily as the requirements evolve during the iterations. This results in planned deliverable products at the end of each iteration to internal and external customers.